Decentralized Reactive Collision Avoidance for Multivehicle Systems

Emmett Lalish and Kristi A. Morgansen

Abstract—This paper addresses a novel approach to the n-vehicle collision avoidance problem. The vehicle model used is a planar unicycle, chosen for its wide applicability to ground, sea, and air vehicles. This paper generalizes previous work with constant-speed vehicles to models that include the ability to slow down, stop and reverse. An algorithm is developed that works in conjunction with any desired controller to guarantee all vehicles remain free of collisions while attempting to follow their desired control. This algorithm is reactive and decentralized, making it well suited for real time applications, and explicitly accounts for actuation limits. Results are demonstrated in simulation.

I. INTRODUCTION

As multi-vehicle autonomous systems are studied and implemented, the issue of conflict resolution becomes an increasingly important point. From mobile robots performing a cooperative search to air traffic control for unmanned aerial vehicles (UAVs), collision avoidance is of utmost importance for safety.

The Decentralized Reactive Collision Avoidance (DRCA) algorithm presented here is an extension of the previous work in [1], which performed collision avoidance for a homogeneous group of constant-speed vehicles. The constant-speed unicycle was chosen for its applicability to aircraft, whose flight envelopes are often so restricted that a constant-speed model is desired.

In order to capture the essential dynamics of a wider range of vehicles, the more general model of a variable-speed unicycle was chosen for this work. However, arbitrary speed restrictions are still modeled, making constant-speed vehicles a subset of this more general framework. The vehicles modeled can now range from surface vehicles to submarines, ships and aircraft. Even mobile robot applications were considered by allowing the speed to be negative, so that the vehicles can reverse.

A useful overview of papers on deconfliction can be found in [2]. The authors divide autonomous conflict resolution methods into three categories: prescribed, optimized, and force field.

Prescribed maneuvers include approaches like [3], [4], [5], where all vehicles follow a set protocol, not unlike the rules of the road. While this approach can lead to straightforward proofs, it also tends to be less flexible with respect to changing conditions.

Optimization schemes are also quite common [6], [7], [8], but suffer in real time applications from non-deterministic computation time. Additionally, these approaches tend to be centralized, which often limit their applicability in real systems.

The DRCA algorithm fits most closely into the force field category, though it differs widely from most other algorithms which use force fields or potential functions. Most force field approaches treat each vehicle as a charged particle that repels all the other vehicles, based primarily on position information (a zeroth order look ahead) [9], [10], [11], [12]. The force field defined in the work here differs in that it makes use of the collision cone concept, which is common in the deconfliction literature [6], [13], [14]. This method involves a first order look ahead for detecting conflicts, which takes the restrictions of the unicycle model into account directly. An implicit assumption is that no antagonistic vehicles are present in the system; either all vehicles are trying to avoid conflicts, or at worst some are maintaining constant velocity.

One caveat of the DRCA algorithm is that it can only keep a system conflict-free. When vehicles start on a collision course, an initial deconfliction maneuver is required to bring the system to a safe state where the DRCA algorithm can take over. One such maneuver is discussed in this paper, and an initial separation bound is presented for which collision avoidance is guaranteed.

The DRCA algorithm is currently planar for applicability to the widest range of systems. Even aircraft are often restricted to a fixed altitude for air traffic control. Also, adding a third dimension actually makes the deconfliction problem easier because it adds another degree of freedom for re-routing. Therefore, the DRCA algorithm can be fairly easily extended to three dimensions.

There are no requirements for homogeneity among the vehicles that make up the system. The DRCA algorithm allows each vehicle to have different size, speed, actuation limitations, and gains. The vehicles can even have completely different tasks they are performing. This framework also implicitly allows static obstacles to be avoided, since they can be modeled as zero-speed vehicles.

The authors of [2] also make a distinction between pairwise and global conflict resolution maneuvers. While the collision cone is fundamentally a pairwise conflict detection scheme, the DRCA algorithm takes into account all of the other vehicles in order to compute the control, making it a global approach.

As the name implies, the DRCA algorithm is also decentralized, in that no communication or agreement is required between the vehicles. Each vehicle does require the states of every other vehicle, but this information can come equally from sensing (e.g. radar) as from communication. If commu-
The constraint set \( C \) includes the origin (i.e., the vehicle must always be capable of avoiding collision while simultaneously staying close to the desired communication route). This desired route is the chosen route, the required \( n \) to \( n \) topology is relatively easy to implement through a broadcast. The effect of limited sensor or communication range on this system is a subject of current research and is beyond the scope of this paper.

This paper is organized as follows. Section II gives the problem statement and introduces definitions and notation used throughout the paper. Initial deconfliction maneuvers are discussed in Section III. The DRCA algorithm is described in Section IV. Simulations and discussions of performance are given in Section V. Conclusions and future work are in Section VI.

II. PROBLEM STATEMENT

This paper presents a method for deconflicting \( n \) unicycle vehicles. Each vehicle has a desired control input, \( u_d(t) \), which comes from an arbitrary outer-loop controller. This controller causes the vehicle to perform a desired task, which could be anything, e.g., target tracking, waypoint navigation, area searching, etc. The goal of the DRCA algorithm is to adjust the control input on each vehicle to guarantee collision avoidance while simultaneously staying close to the desired control input (keeping in mind that this desired control will change with time).

The dynamics of the \( i \)th vehicle are:

\[
\frac{d}{dt} \begin{bmatrix} x_i \\ y_i \\ \psi_i \\ u_i \\ u_a \end{bmatrix} = \begin{bmatrix} s_i \cos(\psi_i) \\ s_i \sin(\psi_i) \\ u_i \\ u_a \end{bmatrix},
\]

(1)

where the inputs are heading rate (turning) and forward acceleration:

\[
u_i = \begin{bmatrix} u_i \\ u_a \end{bmatrix}.
\]

The speed of the vehicle \( s_i \) is restricted to lie on a closed interval \( S_i \):

\[
S_i = \{ s_i \in \mathbb{R} \mid s_{i,min} \leq s_i \leq s_{i,max} \}.
\]

(3)

The minimum speed can be zero to simulate a vehicle that can stop but cannot reverse, negative for a vehicle that can reverse, or positive for a vehicle with a minimum speed (e.g., aircraft). One can also have \( s_{min} = s_{max} \) for constant-speed applications.

The inputs are also restricted to a constrained domain by the specific limitations of the vehicle being studied:

\[
u_i \in C_i,
\]

(4)

where

\[
C_i = \{ u_i \in \mathbb{R}^2 \mid u_{i,min} \leq u_i \leq u_{i,max}, \quad u_{a,min} \leq u_a \leq u_{a,max} \}. \quad (5)
\]

The constraint set \( C_i \) can vary with time, but must always include the origin (i.e., the vehicle must always be capable of maintaining its current velocity). In fact, speed restrictions can be implicitly modeled by making \( u_{a,max} \to 0 \) as \( s \to s_{max} \), which is an accurate model of most real vehicles.

Additionally, many vehicles are not modeled well by a rectangular \( C \) (i.e., the maximum acceleration and turning are coupled). To handle this case, one can simply use a \( C \) that encloses the real input constraint set, then apply saturation to the input generated by the DRCA algorithm. The guarantees will hold as long as the signs of the inputs are correct (hence the requirement that the input constraint set contain the origin).

A. Definitions

The position vector of vehicle \( i \) is denoted

\[
r_i \equiv \begin{bmatrix} x_i \\ y_i \end{bmatrix}.
\]

The standard rotation matrix \( R(\psi) \) is used:

\[
R(\psi) \equiv \begin{bmatrix} \cos \psi & -\sin \psi \\ \sin \psi & \cos \psi \end{bmatrix},
\]

(7)

and the heading vector \( h \) and normal vector \( g \) of vehicle \( i \) are:

\[
h_i \equiv \begin{bmatrix} \cos \psi_i \\ \sin \psi_i \end{bmatrix} \quad \text{and} \quad g_i \equiv \begin{bmatrix} -\sin \psi_i \\ \cos \psi_i \end{bmatrix}.
\]

(8)

The velocity vector can now be defined as:

\[
v_i \equiv \dot{r}_i = s_i h_i.
\]

(9)

The relative position vector from vehicle \( i \) to vehicle \( j \) is defined as

\[
r_{ij} \equiv r_j - r_i,
\]

(10)

while the relative velocity vector is defined in the opposite sense:

\[
v_{ij} \equiv v_i - v_j.
\]

(11)

Note that these definitions imply the following:

\[
\dot{r}_{ij} = -v_{ij},
\]

(12)

and

\[
v_{ij} = u_i, s_i g_i + u_a i h_i - u_{ij} s_j g_j - u_a j h_j.
\]

(13)

For constant \( v_{ij} \), the relative position evolves as

\[
r_{ij}(t) = -\int_0^t v_{ij} dt = r_{ij}(0) - v_{ij} t,
\]

(14)

where \( t = 0 \) means the current time. To simplify the notation in the rest of this paper, \( t = 0 \) will be assumed and the \( ij \) subscripts will be dropped (for example, \( r_{ij}(0) \) will be written as \( r \)).

In order to avoid collisions, first a strict definition of collision is necessary. These vehicles are modeled as non-holonomic point masses, however real vehicles have finite size. Therefore in order to collide, the vehicles do not have to attain the same position in space at the same time, but rather come within a minimum allowed distance of each other at some point in time. This minimum distance could be, for example, the five nautical mile separation between aircraft required by the FAA or the sum of the radii of two mobile robots.
Definition 1 (Collision): A collision occurs between two vehicles when

\[ \| r \| < d_{sep}, \] (15)

where \( d_{sep} \) is the minimum allowed separation distance between the vehicles’ centers. This distance can be different for each pair of vehicles in order to account for heterogeneity in the system.

For two vehicles not in a collision, the next question is whether they will collide if they remain at their present velocities. This situation will be called a conflict.

Definition 2 (Conflict): A conflict occurs between two vehicles \((i\) and \(j)\) if they are not currently in a collision, but with null control inputs (i.e. constant velocity), will at some future point in time enter a collision:

\[ \min_{t>0} \| r(t) \| < d_{sep}. \] (16)

Lemma 1: A necessary and sufficient condition for there to be no conflict is

\[ |\beta| \geq \alpha, \] (17)

where

\[ \beta = \angle v - \angle r, \] (18)

and

\[ \alpha = \arcsin \left( \frac{d_{sep}}{\| r \|} \right). \] (19)

This lemma is proven in [1]. The angle \( \alpha \) denotes the half-width of the collision cone, similar to [6], [13], [14], and described geometrically in Fig. 1.

III. INITIAL DECONFLICTION MANEUVERS

Section IV contains a proof that the DRCA algorithm can keep a system conflict-free indefinitely once a conflict-free state is achieved. However, generally the point of collision avoidance is to resolve conflicts that are already present. Hence some initial maneuver is required by the vehicles to bring themselves from a conflicted state to a conflict-free state so that the DRCA algorithm can keep them that way.

The goal of the initial deconfliction maneuver is to bring the vehicles to a conflict-free state as quickly as possible and prove that there will be no collisions during this time, given certain bounds on the initial conditions. The bounds are important because a wide class of initial conditions exist for which collision avoidance is impossible (usually due to lack of sufficient control authority).

There are many possible maneuvers for initial deconfliction, but this section will detail the most promising options studied so far. A first consideration on how to maneuver out of conflict is which of the inputs should dominate? Changing speed is a problem for two main reasons. First, most vehicles have less capability of forward acceleration than lateral acceleration (less control authority). Second, all vehicles have speed limits, which means at some point control authority (in one direction) vanishes entirely. Therefore the deconfliction maneuver presented here will amount to a turning-only command.

The simplest maneuver is to have all vehicles turn the same way at maximum rate until a conflict-free state is reached for the whole system. If the direction is pre-programmed (e.g. positive or left), this maneuver mimics a rules of the road approach, where vehicles always pass on the same side. A similar approach was used in [3], because for exact or nearly-exact conflicts, this maneuver results in a roundabout passing behavior, which tends to be desired.

However, more important than performance is a guarantee of safety. Collision avoidance for the “all hard left” maneuver can be proven because even when it is impossible to get the system to a conflict-free state, the maneuver simply becomes a loiter pattern.

Theorem 1: A system of \( n \) vehicles will remain collision free for all time if each vehicle constantly turns at its maximum rate while maintaining speed if the initial separation of each pair of vehicles satisfies

\[ \| r_{ij} \| \geq 2 \frac{s_i}{u_{i,max}} + 2 \frac{s_j}{u_{j,max}} + d_{sep_{ij}}. \] (20)

Proof: The loiter pattern each vehicle makes by constant turning is a circle of radius \( \frac{s_i}{u_{i,max}} \). As long as a pair of vehicles is separated by at least the sum of the diameters of their loiter patterns and the minimum separation distance, then they can never collide.

For initial conditions where the conflicts are inexact (as defined in [3]), this rules of the road approach can occasionally lead to the initial deconfliction maneuver taking longer than necessary (i.e. more control effort and course deviation). Nowhere in the proof does it stipulate that each vehicle must turn the same way, only that each must pick a direction and stick with it. It seems reasonable therefore that some kind of heuristic or even an optimization could be capable of giving better performance than the simple “all hard left” command. This is a topic of current research. Additionally, application-specific adjustments could include changing speed to attain the minimum vehicle turning radius.
IV. DRCA ALGORITHM

Once the initial deconfliction maneuver has been performed and the system is in a conflict-free state, the DRCA algorithm takes over and allows each vehicle to use its desired control input unless that input would cause it to come into conflict with another vehicle.

In order to smoothly transition from the desired control to the avoidance control, it is necessary to have a measure defining how close to a conflict a particular pair of vehicles are. To define this measure, first find the vector \( c \) defining the near side of the collision cone:

\[
 c = R(\text{sgn}(\beta) \alpha)^T \frac{r}{\|r\|},
\]

where \( \text{sgn}(x) \) is the sign function, taking on the value 1 for \( x \geq 0 \) and -1 for \( x < 0 \).

Next, construct the normal vector \( n \) from the collision cone to the relative velocity vector (see Fig. 2). If \( c^T v \leq 0 \) (the vehicles are headed away from each other) then the nearest point on the collision cone is the tip and \( n = v \). Otherwise it can be found from the following two geometric relations:

\[
 c^T n = 0 \\
 v_i = v + kc + n,
\]

which can be condensed into a single matrix equation

\[
 \begin{bmatrix}
 I & c \\
 c^T & 0
\end{bmatrix}
\begin{bmatrix}
 n \\
 k
\end{bmatrix} =
\begin{bmatrix}
 v \\
 0
\end{bmatrix},
\]

and solved for \( n \) (\( k \) is not used), yielding the second half of the piecewise definition:

\[
 n = \begin{cases}
 v, & e^T v \leq 0 \\
 R \left( \frac{z}{z} \right) c^T cR \left( \frac{z}{z} \right)^T v, & e^T v > 0.
\end{cases}
\]

Because the control inputs \( u_a \) and \( u_t \) produce changes in velocity along the orthogonal vectors \( g \) and \( h \) respectively (13), they can be decoupled and analyzed separately. Therefore a measure is needed to tell the distance in each of these coordinate directions from the collision cone. Define the following measures (valid only when not in conflict):

\[
 p_t = \frac{\|s\|^2}{s_i n^T g_i} \quad \text{and} \quad p_a = \frac{\|n\|^2}{n^T h_i}.
\]

The geometry of these measures is shown in Fig. 2 for \( e^T v > 0 \). The intuition behind these measures is that \( p_a \) is the amount of forward velocity change that would result in a conflict, while \( p_t \) is a linear approximation of the amount of heading change that would result in a conflict, valid for \( |\beta| - \alpha \ll 1 \) (small angle approximation).

For \( c^T v < 0 \), the only difference is that the \( p \)'s are measured to the half-space perpendicular to \( n \) (which must by definition enclose the collision cone), rather than to the collision cone itself. Effectively the measures are conservative in this case, but in this way they retain continuity across the domain. Both \( p \)'s are signed such that positive means a positive control input will push the vehicle farther from the conflict and negative means a negative control input will push the vehicle farther from the conflict.

If the system is conflict-free and a vehicle has no nearby conflicts, then it will simply follow its desired control. Define \( \epsilon_t, \epsilon_a > 0 \) to represent how large \( p_t \) and \( p_a \) should be to ignore a conflict.

The \( n \)-vehicle DRCA algorithm running on vehicle \( i \) computes and finds the closest conflicts, i.e.

\[
 p_t^+ = \min \{ p_t > 0, \epsilon_t \} \\
 p_t^- = -\max \{ p_t < 0, -\epsilon_t \},
\]

and likewise for \( p_a \). Note that by definition \( 0 < p_t^\pm \leq \epsilon \). For any equation in this section that has no \( a \) or \( t \) subscripts, it can be assumed that the equation holds in both the acceleration and turning directions.

Each control input is found using the control function, \( F \):

\[
 u_t = F(p_t^+, p_t^-) \\
 u_a = F(p_a^+, p_a^-).
\]

The control function chosen for this implementation of the DRCA algorithm is piecewise-linear, defined by the following ordered triples of the form \( (p^+, p^-, u) \):

\[
 P_1 = (0, 0, 0) \\
 P_2 = (\epsilon, 0, u_{\text{max}}) \\
 P_3 = (0, \epsilon, u_{\text{min}}) \\
 P_4 = (\epsilon, \epsilon, u_d),
\]

where \( P_1, P_2, P_3 \) define one plane and \( P_2, P_3, P_4 \) define the other. An example of this control function is shown in Fig. 3. Because \( F \) is a function of the desired control, \( u_d \) must be saturated such that

\[
 u_{\text{min}} \leq u_d \leq u_{\text{max}}.
\]

A more clear way to choose the \( \epsilon \)'s is to relate them to gain-like parameters,

\[
 k_t = \frac{u_{t,\text{max}} - u_{t,\text{min}}}{\epsilon_t}
\]

(and similarly for \( k_a \)). Now both \( k_t \) and \( k_a \) have units of inverse seconds. Also, it can be seen that the magnitude of the gradient of the control function will always be less than or equal to \( k \), depending on the desired control.
Theorem 2: The DRCA algorithm described above, when implemented on $n$ vehicles with dynamics (1) and input constrained by (4), will keep the system collision-free for all time if the system starts conflict-free.

Proof: To measure the distance to a collision, define $m$ as a signed version of $\|n\|$:

$$m = \begin{cases} \|v\|, & c^Tv \leq 0 \\ \|v\| \sin(|\beta| - \alpha), & c^Tv > 0. \end{cases}$$

Note that $m$ is negative during conflict and positive during no conflict.

To ensure that a conflicted state is never reached, it is sufficient to show that

$$\lim_{m \to 0} \dot{m} \geq 0. \quad (32)$$

For $c^Tv \leq 0$ (using the $ij$ notation again briefly for clarity),

$$\dot{m}_{ij} = \frac{n_i^T \dot{v}_{ij}}{m}$$

$$= \frac{1}{m} \left( u_i s_i n_{ij}^T g_i + u_{ai} n_{ij}^T h_i \right)$$

$$- u_i s_j n_{ij}^T g_j - u_{ai} n_{ij}^T h_j, \quad (33)$$

and because of the symmetry of the problem $n_{ji} = -n_{ij}$, and so

$$\dot{m}_{ij} = \frac{1}{m} \left( u_i s_i n_{ij}^T g_i + u_{ai} n_{ij}^T h_i \right)$$

$$+ u_i s_j n_{ij}^T g_j + u_{ai} n_{ij}^T h_j$$

$$= \frac{\|n\|^2}{m} \left( \frac{u_i}{p_{i,ij}} + \frac{u_{ai}}{p_{ai,ij}} + \frac{u_i}{p_{i,ji}} + \frac{u_{aj}}{p_{a,ji}} \right). \quad (34)$$

As long as the controller ensures that each $u_i$ has the same sign as its $p_i$ and each $u_{ai}$ has the same sign as its $p_{ai}$, then it can be seen that $\dot{m} \geq 0$ for that pair of vehicles. Note that each vehicle need only calculate its control from its own point of view, and that this rule will automatically make the vehicles cooperate in avoiding conflicts.

For $c^Tv > 0$, the derivative of (31) becomes

$$\dot{m} = \sin(|\beta| - \alpha) \frac{d\|v\|}{dt} + \|v\| \cos(|\beta| - \alpha) \frac{d|\beta|}{dt}. \quad (35)$$

From the geometry,

$$\frac{d|\beta|}{dt} = \text{sgn}(\beta) \left( \frac{d\|v\|}{dt} - \frac{d\|r\|}{dt} \right)$$

$$= \text{sgn}(\beta) \frac{d\|v\|}{dt} + \frac{\|v\|}{\|r\|} |\sin \beta| \quad (36)$$

From [1],

$$\frac{d}{dt} = \frac{\|v\|}{\|r\|} \cos \beta \tan \alpha. \quad (37)$$

Now one has

$$\frac{d}{dt} (|\beta| - \alpha) = \frac{\|v\|}{\|r\|} (|\sin \beta| - \cos \beta \tan \alpha) + \text{sgn}(\beta) \frac{d\|v\|}{dt}$$

which can be substituted into (35) to form

$$\dot{m} = \sin(|\beta| - \alpha) \frac{d\|v\|}{dt} + \cos(|\beta| - \alpha) \text{sgn}(\beta) \|v\| \frac{d\|v\|}{dt}$$

$$+ \cos(|\beta| - \alpha) \|v\|^2 \left( |\sin \beta| - \cos \beta \tan \alpha \right). \quad (38)$$

For $c^Tv > 0$,

$$n^T \dot{v} = m \left( \sin(|\beta| - \alpha) \frac{d\|v\|}{dt} \right)$$

$$+ \cos(|\beta| - \alpha) \text{sgn}(\beta) \|v\| \frac{d\|v\|}{dt} \right). \quad (39)$$

Therefore (38) reduces to

$$\dot{m} = \frac{n^T \dot{v}}{m} + \cos(|\beta| - \alpha) \frac{\|v\|^2}{\|r\|} \left( |\sin \beta| - \cos \beta \tan \alpha \right). \quad (40)$$

Assuming the system is conflict-free, Lemma 1 dictates $|\beta| \geq \alpha$. Therefore $|\tan \beta| \geq \tan \alpha$, and so

$$|\sin \beta| - \cos \beta \tan \alpha \geq 0. \quad (41)$$

Also, $c^Tv > 0$ implies that $\cos(|\beta| - \alpha) > 0$. Therefore, positivity for $\dot{m}$ reduces to the same conditions on $u_i$ and $u_{ai}$ as found from (34).

Combining this result with the definitions (26), any continuous control function that ensures for all vehicles

$$\lim_{p_i \to 0} u_i \geq 0, \quad \lim_{p_i \to 0} u_i \leq 0, \quad \lim_{p_i \to 0} u_{ai} \geq 0, \quad \lim_{p_i \to 0} u_{ai} \leq 0, \quad \lim_{p_i \to 0} u_{ai} \geq 0, \quad \lim_{p_i \to 0} u_{ai} \leq 0, \quad (42)$$

also ensures that

$$\lim_{m \to 0} \dot{m} \geq 0, \quad (43)$$

which means the system cannot enter a conflicted state.

The control function used in this implementation (28) satisfies (42) and so the DRCA algorithm will cause the $n$-vehicle system to remain conflict-free for all time, assuming it started that way. Note this result holds for arbitrary (even time varying) $u_d, u_{min}$ and $u_{max}$, so long as they satisfy (29) and $\mathcal{C}$ contains the origin at every instant. $lacksquare$
V. SIMULATIONS

A. Variable Speed

The first simulation (Fig. 4) is of a homogeneous group of five vehicles that are capable of moving both forward and backward. Each vehicle has a target that moves at constant velocity, whose position and orientation the vehicle is attempting to attain (using a standard target following controller). The vehicles are all initialized with random headings and negative initial speeds (see Fig. 5), so the corners in the paths denote points where the vehicle speed crosses zero. The initial positions of the vehicles form a circle of radius six meters, with their targets heading toward the center (with some randomization). The black dotted line denotes the speed of the targets.

These vehicles start conflict-free, so no deconfliction maneuver is necessary. However, the targets move in such a way that following the desired controls would cause the vehicles to collide. The DRCA algorithm successfully keeps the system conflict-free while eventually allowing the vehicles to attain their targets. The turning rate control for each vehicle is shown in Fig. 6. To describe how the controller operates, Fig. 7 shows the collision cones from the red vehicle’s perspective at the same instant that the vehicles are plotted in Fig. 4. Interesting to note is that while the blue and purple vehicles are the nearest in distance to the red vehicle, it is the purple and yellow vehicles that are closest to conflict.

B. Heterogeneity

The second simulation is geared more toward aircraft, as the group of five vehicles in Fig. 8 are restricted to a constant speed of 1 m/s. In addition to the five vehicles is one static obstacle (which can be thought of as a zero speed vehicle). In this simulation, $d_{sep}$ is the sum of the radii of the two vehicles involved, and the shaded regions represent the radius of each vehicle. The vehicles’ initial conditions form them into a circle of radius eight meters and point them roughly toward the obstacle, so the system starts in conflict.

The vehicles perform the “all turn left” initial deconfliction maneuver (as seen in Fig. 9), which brings the system to a
VI. CONCLUSION

This work has developed a decentralized control algorithm for deconflicting \( n \) unicycle-type vehicles. The DRCA algorithm is reactive and so can easily be implemented real time on a wide variety of vehicles, including aircraft, ships, submarines and cars. Collision avoidance is guaranteed for a general \( n \)-vehicle system once a conflict-free state is reached, even in the case of arbitrarily small control authority. A lower bound was found for the initial separation of the \( n \) vehicles such that collision avoidance is assured even when starting in conflict. Finally, the DRCA algorithm allows the vehicles to follow changing desired controls so long as safety is not sacrificed.

Many extensions are possible for this work. First, for applications to aircraft and submarines, extending this concept to three dimensions will add a degree of freedom and hence increase the performance of the system. Second, the DRCA algorithm shows promise for evading adversarial pursuers, and may lead to interesting results regarding pursuit/evasion games. Third, finding better performing initial deconfliction maneuvers will make this algorithm more suitable to general conflict scenarios. Finally, integrating a finite detection horizon into this algorithm will be important to maintain the collision avoidance guarantees whilst operating in a real environment where dangers are not known until they are close enough. These areas are currently under investigation.

REFERENCES